Overview

Goal
- Multi-task Learning with ConvNets
- No exhaustive search across architectures

Key Ideas
- Learn information sharing across tasks
- Explores exponentially more architectures in a principled manner
- Generalizes across tasks

Object
- Explores exponentially more architectures in a principled manner
- Generalizes across tasks

Standard Approach to Multi-task Learning

All shared layers
- Input
- Task A
- Task B

More task-specific layers
- Input
- Task A
- Task B

Spectrum of sharing: Exploring all ‘split’ Architectures

Relative performance wrt. Specific Network
- Attributes Classification (mAP)
- Object Detection (mAP)

Problems and Limitations
- Practically Expensive
- No principled way of exploring architectures
- One architecture may not improve all tasks
- Does not generalize across tasks

Approach

Cross-stitching

- Two networks trained independently
- Combine information across layers and networks
- Explores all split architectures and more
- Performs better than the best ‘split’ architecture

Cross-stitch unit: Learning to share

- Model information sharing as a linear combination
- Fuse information from multiple tasks
- Unit decides how much information to share

How to Learn?

- Initialize unit learning rates to be higher than layer learning rates
- Better to initialize two networks with “task-specific” networks
- Units are not too sensitive to initial $\alpha$ values
- Adds few parameters (maximum of one scalar per filter channel)
- Enforcing convexity in units does not help

What do these units learn? Visualizing and more.

Surface Normal and Semantic Segmentation on NYUv2

- More sharing at lower layers – pool1, pool2
- Less sharing at higher layers – fc6, fc7
- Least sharing at pool5
- Similar trend at different initializations

Classes with less data

Sort classes according to no. of instances, and plot change in performance

Attributes from PASCAL VOC 2008

Object Detection and Attributes on PASCAL VOC 2008

<table>
<thead>
<tr>
<th>Method</th>
<th>Detection mAP</th>
<th>Attributes mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Task Network</td>
<td>44.9</td>
<td>60.9</td>
</tr>
<tr>
<td>MTL Baseline (Zhou et al., 2013)</td>
<td>42.7</td>
<td>54.1</td>
</tr>
<tr>
<td>Ensemble of two Networks</td>
<td>46.1</td>
<td>61.1</td>
</tr>
<tr>
<td>Split architecture (brute-force) Attributes</td>
<td>44.6</td>
<td>61.0</td>
</tr>
<tr>
<td>Split architecture (brute-force) Detection</td>
<td>44.8</td>
<td>59.7</td>
</tr>
<tr>
<td>Cross-stitch (Ours)</td>
<td>45.2</td>
<td>63.0</td>
</tr>
</tbody>
</table>

Surface Normal and Semantic Segmentation on NYUv2

<table>
<thead>
<tr>
<th>Method</th>
<th>Surface Normal Median Error</th>
<th>Semantic Segmentation mIU</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Task Network</td>
<td>19.0</td>
<td>18.4</td>
</tr>
<tr>
<td>MTL Baseline (Zhou et al., 2013)</td>
<td>18.9</td>
<td>16.6</td>
</tr>
<tr>
<td>Ensemble of two Networks</td>
<td>18.5</td>
<td>18.9</td>
</tr>
<tr>
<td>Split architecture (brute-force)</td>
<td>19.1</td>
<td>19.2</td>
</tr>
<tr>
<td>Cross-stitch (Ours)</td>
<td>18.2</td>
<td>19.3</td>
</tr>
</tbody>
</table>

Cross-stitch unit

- $\alpha_0$ = 0.9, $\alpha_5$ = 0.1
- $\alpha_5$ = 0.5, $\alpha_7$ = 0.5
- $\alpha_0$ = 0.1, $\alpha_7$ = 0.9

Learned cross-stitch units

SN  Seg
- More sharing at lower layers – pool1, pool2
- Less sharing at higher layers – fc6, fc7
- Least sharing at pool5
- Similar trend at different initializations